Support of Language-Specific Characters

This section describes how Natural supports language-specific characters and covers the following topics:

- Why is the Support of Language-Specific Characters Important?
- The Configuration File NATCONV.INI
- Sample NATCONV.INI File

Why is the Support of Language-Specific Characters Important?

The support of language-specific characters can help you when using:

- Upper-/Lower-Case Conversion
- Identifier Checking
- Character Classification
- Upper-/Lower-Case Translation of language-specific characters;
- language-specific characters in Natural identifiers, object names and library names;
- language-specific characters in an operand compared with a mask definition
  (see **MASK Option** in the Natural Reference documentation).

Upper-/Lower-Case Conversion

Natural performs upper-/lower-case conversion if you specify one of the following items:

- the "%U" terminal command,
- the "AD=T" field attribute,
- the EXAMINE TRANSLATE statement.

By modifying the translation tables, you can support language-specific characters.

Identifier Checking

Natural checks identifiers (that is, user-defined variables in source programs), names of Natural objects and names of Natural libraries by using check tables. By modifying these tables, you can, for example, allow language-specific characters. In addition, you can redefine the characters "+", "#" and "&", which have a special meaning when used as the first character in variable names.

Character Classification

Natural performs an internal character classification when comparing an operand with a MASK option, and for the default delimiter used in the EXAMINE and SEPARATE statements. This classification is done via default tables. By modifying these tables, you can support language-specific characters.

Configuration File NATCONV.INI

All check, translation and classification tables used by Natural to support language-specific characters reside in the configuration file NATCONV.INI.

You can modify NATCONV.INI to support local or application-specific characters.

In a standard application, NATCONV.INI need not and should not be modified, because this could lead to serious inconsistencies, in particular if Natural objects and database data are already present.

Any modifications of NATCONV.INI should be well considered and carefully performed, otherwise problems might occur that are difficult to locate.

NATCONV.INI is subdivided in sections and subsections. It contains four defined sections called:

- CHARACTERSET-DEFINITION
- CASE-TRANSLATION
- IDENTIFIER-VALIDATION
- CHARACTER-CLASSIFICATION

CHARACTERSET-DEFINITION	This section defines the name of the internal character set. The default is ISO8859_1. If you choose a different character set, subsections for this character set must be contained in the following sections.
CASE-TRANSLATION	This section contains the tables required for the conversion from upper to lower case. This conversion is done within the internal character set. If, for example, the internal character set is "ISO8859_5", the following two subsections must be contained in this section: - [ISO8859_5->UPPER] - [ISO8859_5->LOWER]
IDENTIFIER-VALIDATION	This section contains the tables required for the validation of identifiers, object names and library names. It contains a subsection for each defined internal character set. The special characters "#" (for non-database variables), "+" (for application-independent variables), "@" (for SQL and Adabas null or length indicators) and "&" (for dynamic source generation) can be redefined in this section. In addition, the set of valid first and subsequent characters for identifiers, object names and library names can be modified.
CHARACTER-CLASSIFICATION	This section contains the tables required for the classification of characters, which, for example, are used when evaluating the MASK option. It contains a subsection for each defined internal character set.

The section CHARACTERSET-DEFINITION as well as each subsection contain lines which describe how characters are to be converted and which characters are related with which attributes.

These lines are represented as follows:

```
line ::= key
= value key ::= name_key | range_key name_key
::= keyword{ CHARS } keyword ::= INTERNAL-CHARACTERSET | NON-DB-VARI
| DYNAMIC-SOURCE | GLOBAL-VARI | FIRST-CHAR | SUBSEQUENT-CHAR | LIB-FIRST-CHAR
| LIB-SUBSEQUENT-CHAR | ALTERNATE-CARET ISASCII | ISALPHA | ISALNUM | ISDIGIT
| ISXDIGIT | ISLOWER | ISUPPER | ISCNTRL | ISPRINT | ISPUNCT | ISGRAPH | ISSPACE
range_key ::= hexnum | hexnum-hexnum value
::= val {, val } val ::= hexnum | hexnum-hexnum
hexnum ::= xhexdigithexdigit | Xhexdigithexdigit
```

Notes:
If "range_key" variable is specified on the left-hand side, the number of values specified on the right-hand side must correspond to the number of values specified in the key range, except if only one value is specified on the right-hand side, which is then assigned to each element of the key range.

When the "name_key" variable is specified on the left-hand side and the corresponding list of character codes does not fit in one line, it can be continued on the next line by specifying"name_key =" again.

Examples for Valid Lines:

x00-x1f = x00	All characters between "x00" and "x1f" are converted to "x00".
x00-x7f = x00-x7f	All characters between "x00" and "x7f" are not converted.
x00-x08 = x00,x01-x07,x00	The characters "x00" and "x08" are converted to "x00" and characters between "x01" and "x07" are not converted.

ISALPHA = x41-x5a,x61-x7a,xc0-xd6,xd8
ISALPHA = xd9-xf6,xf8-xff
	The attribute ISALPHA is assigned to all
	characters specified in these two lines.

Examples for Invalid Lines:

x41 = 'A'	All characters must be specified in hexadecimal format.
0x00-0x1f = 0x00	Hexadecimal values have to be specified in either of the following ways: x*digitdigit* X*digitdigit*
x00-x0f = x00,x01	The number of specified values does not correspond to the number of elements in the key range.

Sample NATCONV.INI File

```
#************************************************************************
# Natural character set configuration file #************************************************************************
# # NATCONV.INI consists of five sections: # # CHARACTERSET-DEFINITION - Defines
the name of the character set that is to # be used by Natural. This name has to
appear in # the subsection names of all following sections. # IBMCP_850 (PC) or
ISO8859_1 (UNIX) are defined # as default character sets. # # CHARACTERSET-TRANSLATION
- In addition to the character set used by Natural # (the so-called internal character
set) there may # exist several external character sets used for # different terminals
connected to the same # Natural. If you want to use a terminal with a # character
set differing from the internal # character set, you have to define a TCS entry
for # the terminal in SAGtermcap (e.g. ":TCS=USASCII:") # and corresponding
subsections (e.g. # "[ISO8859_1->USASCII]" and # "[USASCII->ISO8859_1]")
in this section. # The conversion between external and internal # character set
is performed during terminal I/O. # Note that this conversion is only done if
you are # running Natural for UNIX. # Beside conversion tables for the terminal
# character set there exist additional tables # for special purposes (e.g. tables
for conversion # between ASCII and EBCDIC code). # # CASE-TRANSLATION - This section
defines the translation tables # for lower- and upper-case translation. The #
tables are valid for the English language. If you # want to support language-specific
characters, see # explanation below. #
```

```
# IDENTIFIER-VALIDATION
- This section defines tables for the validation of # Natural identifiers, object
names and library # names. The tables allow identifiers, object # and library
names as described in the Natural # manuals. If you want to support language-specific
# characters, see explanation below. # # CHARACTER-CLASSIFICATION - This section
defines tables for the # classification of characters according to the # corresponding
C library functions (e.g. # isalpha). The tables are valid for the English # language.
If you want to support # language-specific characters, see explanation # below.
# # # How to support language-specific characters with NATCONV.INI: # # As mentioned
above NATCONV.INI supports the English language per # default. If your language
uses additional characters you have to make # some modifications. As an example
the required modifications for the # German language are given as comments in
the corresponding sections. # Note that those modifications might affect the portability
of Natural # source files and/or Natural applications. # # Required modifications:
# # Section CASE-TRANSLATION # Modify the tables for lower- and upper-case translation
so that all your # alphabetic characters are translated correctly. These translations
```

```
must # be consistent with the ISLOWER and ISUPPER tables in the section # CHARACTER-CLASSIFICATION.
# # Section IDENTIFIER-VALIDATION # If you want to allow/forbid special characters
in Natural identifiers, # object or library names, you have to modify the corresponding
tables # (see section IDENTIFIER-VALIDATION). Note that it is not possible to
# define characters as valid identifier characters which have a special # meaning
in the Natural language; e.g. "(". #

# Section
CHARACTER-CLASSIFICATION # Add your alphabetic characters to the tables ISALPHA,
ISALNUM, ISLOWER # (if lower-case), ISUPPER (if upper-case) and ISPRINT. # Beware
that ISLOWER and ISUPPER must be consistent with the translation # tables in the
section CASE-TRANSLATION. # Make sure that no alphabetic character is included
in the tables # ISCNTRL and ISPUNCT. # #************************************************************************

[CHARACTERSET-DEFINITION]

# defining
the internal character set for Natural

INTERNAL-CHARACTERSET
= ISO8859_1

[CHARACTERSET-DEFINITION-END] #**********************************************************************

#************************************************************************
[CHARACTERSET-TRANSLATION]

# translation tables between
internal and external character set

#-----------------------------------------------------------------------
[ISO8859_1->USASCII] #translate ISO8859-1 to USASCII code

x00-x7F = x00-x7F x80-xBF = x3F xC0-xCF = x41,x41,x41,x41,x41,x41,x41,x43,x45,x45,x45,x45,x49,x49,x49,x49
xD0-xDF = x44,x4E,x4F,x4F,x4F,x4F,x4F,x3F,x4F,x55,x55,x55,x55,x59,x50,x73 xE0-xEF
= x61,x61,x61,x61,x61,x61,x61,x63,x65,x65,x65,x65,x69,x69,x69,x69 xF0-xFF = x64,x6E,x6F,x6F,x6F,x6F,x6F,x3F,x6F,x75,x75,x75,x75,x79,x70,x59

[ISO8859_1->USASCII-END] #-----------------------------------------------------------------------

#-----------------------------------------------------------------------
[USASCII->ISO8859_1] #translate USASCII to ISO8859-1 code


x00-xFF = x00-xFF

[USASCII->ISO8859_1-END] #-----------------------------------------------------------------------

#-----------------------------------------------------------------------
[ISO8859_1->ASCII_GERMAN] #translate ISO8859-1 to ASCII_GERMAN code

x00-x7F = x00-x7F x80-xA6 = x3F xA7 = x40 xA8-xBF = x3F xC0-xCF = x41,x41,x41,x41,x5B,x41,x41,x43,x45,x45,x45,x45,x49,x49,x49,x49
xD0-xDF = x44,x4E,x4F,x4F,x4F,x4F,x5C,x3F,x4F,x55,x55,x55,x5D,x59,x50,x7E xE0-xEF
= x61,x61,x61,x61,x7B,x61,x61,x63,x65,x65,x65,x65,x69,x69,x69,x69 xF0-xFF = x64,x6E,x6F,x6F,x6F,x6F,x7C,x3F,x6F,x75,x75,x75,x7D,x79,x70,x59

[ISO8859_1->ASCII_GERMAN-END] #------------------------------------------------------------------

#-----------------------------------------------------------------------
[ASCII_GERMAN->ISO8859_1] #translate ASCII_GERMAN to ISO8859-1 code


x00-xFF = x00-xFF x40 = xA7 x5B-x5D = xC4,xD6,xDC x7B-x7E = xE4,xF6,xFC,xDF

[ASCII_GERMAN->ISO8859_1-END] #------------------------------------------------------------------

# translation tables between internal character set and EBCDIC
code
```

```
#------------------------------------------------------------------------
[ISO8859_1->EBCDIC] #translate ISO8859-1 to EBCDIC code #characters which cannot
be translated properly are replaced by a ’?’

x00-x0F
= x00-x03,x37,x2D,x2E,x2F,x16,x05,x25,x0B-x0F x10-x1F = x10-x13,x3C,x3D,x32,x26,x18,x19,x3F,x27,x1C-x1F
x20-x2F = x40,x5A,x7F,x7B,x5B,x6C,x50,x7D,x4D,x5D,x5C,x4E,x6B,x60,x4B,x61 x30-x3F
= xF0-xF9,x7A,x5E,x4C,x7E,x6E,x6F x40-x4F = x7C,xC1-xC9,xD1-xD6 x50-x5F = xD7-xD9,xE2-xE9,xAD,xE0,xBD,x6A,x6D
x60-x6F = x79,x81-x89,x91-x96 x70-x7F = x97-x99,xA2-xA9,x8B,x4F,x9B,xA1,x07 x80-xFF
= x6F

[ISO8859_1->EBCDIC-END] #--------------------------------------------------------------------------------------

#------------------------------------------------------------------------
[EBCDIC->ISO8859_1] #translate EBCDIC to ISO8859-1 code #characters which cannot
be translated properly are replaced by a ’?’

x00-x0F
= x00-x03,x3F,x09,x3F,x7F,x3F,x3F,x3F,x0B-x0F x10-x1F = x10-x13,x3F,x3F,x08,x3F,x18,x19,x1A,x3F,x1C-x1F
x20-x2F = x3F,x3F,x1C,x3F,x3F,x0A,x17,x1B,x3F,x3F,x3F,x3F,x05,x06,x07,x3F x30-x3F
= x3F,x3F,x3F,x3F,x3F,x3F,x3F,x04,x3F,x3F,x3F,x3F,x14,x15,x3F,x3F x40-x4F = x20,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x2E,x3C,x28,x2B,x7C
x50-x5F = x26,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x21,x24,x2A,x29,x3B,x5E x60-x6F
= x2D,x2F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x2C,x25,x5F,x3E,x3F x70-x7F = x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3A,x23,x40,x27,x3D,x22
x80-x8F = x3F,x61-x69,x3F,x7B,x3F,x3F,x3F,x2B x90-x9F = x3F,x6A-x72,x3F,x7D,x3F,x3F,x3F,x3F
xA0-xAF = x3F,x3F,x73-x7A,x3F,x3F,x3F,x3F,x3F,x3F xB0-xBF = x3F xC0-xCF = x3F,x41-x49,x3F,x3F,x3F,x3F,x3F,x3F
xD0-xDF = x3F,x4A-x52,x3F,x3F,x3F,x3F,x3F,x3F xE0-xEF = x3F,x3F,x53-x5A,x3F,x3F,x3F,x3F,x3F,x3F
xF0-xFF = x30-x39,x3F,x3F,x3F,x3F,x3F,x3F

[EBCDIC->ISO8859_1-END]
#------------------------------------------------------------------------

#------------------------------------------------------------------------
[IBMCP_850->EBCDIC] #translate IBMCP_850 to EBCDIC code #characters which cannot
be translated properly are replaced by a ’?’

x00-x0F
= x00-x03,x37,x2D,x2E,x2F,x16,x05,x25,x0B-x0F x10-x1F = x10-x13,x3C,x3D,x32,x26,x18,x19,x3F,x27,x1C-x1F
x20-x2F = x40,x5A,x7F,x7B,x5B,x6C,x50,x7D,x4D,x5D,x5C,x4E,x6B,x60,x4B,x61 x30-x3F
= xF0-xF9,x7A,x5E,x4C,x7E,x6E,x6F x40-x4F = x7C,xC1-xC9,xD1-xD6 x50-x5F = xD7-xD9,xE2-xE9,xAD,xE0,xBD,x6A,x6D
x60-x6F = x79,x81-x89,x91-x96 x70-x7F = x97-x99,xA2-xA9,x8B,x4F,x9B,xA1,x07 x80-xFF
= x6F

[IBMCP_850->EBCDIC-END] #------------------------------------------------------------------------------------

#------------------------------------------------------------------------
[EBCDIC->IBMCP_850] #translate EBCDIC to IBMCP_850 code #characters which cannot
be translated properly are replaced by a ’?’

x00-x0F
= x00-x03,x3F,x09,x3F,x7F,x3F,x3F,x3F,x0B-x0F x10-x1F = x10-x13,x3F,x3F,x08,x3F,x18,x19,x1A,x3F,x1C-x1F
x20-x2F = x3F,x3F,x1C,x3F,x3F,x0A,x17,x1B,x3F,x3F,x3F,x3F,x05,x06,x07,x3F x30-x3F
= x3F,x3F,x3F,x3F,x3F,x3F,x3F,x04,x3F,x3F,x3F,x3F,x14,x15,x3F,x3F x40-x4F = x20,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x2E,x3C,x28,x2B,x7C
x50-x5F = x26,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x21,x24,x2A,x29,x3B,x5E x60-x6F
= x2D,x2F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x2C,x25,x5F,x3E,x3F x70-x7F = x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3F,x3A,x23,x40,x27,x3D,x22
x80-x8F = x3F,x61-x69,x3F,x7B,x3F,x3F,x3F,x2B x90-x9F = x3F,x6A-x72,x3F,x7D,x3F,x3F,x3F,x3F
xA0-xAF = x3F,x3F,x73-x7A,x3F,x3F,x3F,x3F,x3F,x3F xB0-xBF = x3F xC0-xCF = x3F,x41-x49,x3F,x3F,x3F,x3F,x3F,x3F
xD0-xDF = x3F,x4A-x52,x3F,x3F,x3F,x3F,x3F,x3F xE0-xEF = x3F,x3F,x53-x5A,x3F,x3F,x3F,x3F,x3F,x3F
xF0-xFF = x30-x39,x3F,x3F,x3F,x3F,x3F,x3F

[EBCDIC->IBMCP_850-END]
#------------------------------------------------------------------------

[CHARACTERSET-TRANSLATION-END] #******************************************************************************

#*************************************************************************
[CASE-TRANSLATION]

# translation tables for lower/uppercase
conversion of the internal character set # The translation tables are valid for
the English language. # If you want to handle language specific characters you
have to modify the # following tables accordingly. # The required modifications
for the German language are given as comment.
```

```
#-----------------------------------------------------------------------
[ISO8859_1->LOWER] #translate ISO8859-1 to lowercase


#---- english version x00-x3F = x00-x3F x40-x4F = x40,x61,x62,x63,x64,x65,x66,x67,x68,x69,x6A,x6B,x6C,x6D,x6E,x6F
x50-x5F = x70,x71,x72,x73,x74,x75,x76,x77,x78,x79,x7A,x5B,x5C,x5D,x5E,x5F x60-x7F
= x60-x7F x80-xFF = x80-xFF

#---- german version #x00-x3F
= x00-x3F #x40-x4F = x40,x61,x62,x63,x64,x65,x66,x67,x68,x69,x6A,x6B,x6C,x6D,x6E,x6F
#x50-x5F = x70,x71,x72,x73,x74,x75,x76,x77,x78,x79,x7A,x5B,x5C,x5D,x5E,x5F #x60-xBF
= x60-xBF #xC0-xCF = xC0,xC1,xC2,xC3,xE4,xC5,xC6,xC7,xC8,xC9,xCA,xCB,xCC,xCD,xCE,xCF
#xD0-xDF = xD0,xD1,xD2,xD3,xD4,xD5,xF6,xD7,xD8,xD9,xDA,xDB,xFC,xDD,xDE,xDF #xE0-xFF
= xE0-xFF

[ISO8859_1->LOWER-END] #-------------------------------------------------------------

#-----------------------------------------------------------------------
[ISO8859_1->UPPER] #translate ISO8859-1 to uppercase


#---- english version x00-x5F = x00-x5F x60-x6F = x60,x41,x42,x43,x44,x45,x46,x47,x48,x49,x4A,x4B,x4C,x4D,x4E,x4F
x70-x7F = x50,x51,x52,x53,x54,x55,x56,x57,x58,x59,x5A,x7B,x7C,x7D,x7E,x7F x80-xFF
= x80-xFF

#---- german version #x00-x5F = x00-x5F #x60-x6F
= x60,x41,x42,x43,x44,x45,x46,x47,x48,x49,x4A,x4B,x4C,x4D,x4E,x4F #x70-x7F = x50,x51,x52,x53,x54,x55,x56,x57,x58,x59,x5A,x7B,x7C,x7D,x7E,x7F
#x80-xDF = x80-xDF #xE0-xEF = xE0,xE1,xE2,xE3,xC4,xE5,xE6,xE7,xE8,xE9,xEA,xEB,xEC,xED,xEE,xEF
#xF0-xFF = xF0,xF1,xF2,xF3,xF4,xF5,xD6,xF7,xF8,xF9,xFA,xFB,xDC,xFD,xFE,xFF

[ISO8859_1->UPPER-END] #-------------------------------------------------------------

#-----------------------------------------------------------------------
[IBMCP_850->LOWER] #translate IBMCP_850 to lowercase


#---- english version x00-x3F = x00-x3F x40-x4F = x40,x61,x62,x63,x64,x65,x66,x67,x68,x69,x6A,x6B,x6C,x6D,x6E,x6F
x50-x5F = x70,x71,x72,x73,x74,x75,x76,x77,x78,x79,x7A,x5B,x5C,x5D,x5E,x5F x60-x7F
= x60-x7F x80-xFF = x80-xFF

#---- german version #x00-x3F
= x00-x3F #x40-x4F = x40,x61,x62,x63,x64,x65,x66,x67,x68,x69,x6A,x6B,x6C,x6D,x6E,x6F
#x50-x5F = x70,x71,x72,x73,x74,x75,x76,x77,x78,x79,x7A,x5B,x5C,x5D,x5E,x5F #x60-x7F
= x60-x7F #x80-x8F = x80,x81,x82,x83,x84,x85,x86,x87,x88,x89,x8A,x8B,x8C,x8D,x84,x8F
#x90-x9F = x90,x91,x92,x93,x94,x95,x96,x97,x98,x94,x81,x9B,x9C,x9D,x9E,x9F #xA0-xFF
= xA0-xFF

[IBMCP_850->LOWER-END] #-------------------------------------------------------------

#-----------------------------------------------------------------------
[IBMCP_850->UPPER] #translate IBMCP_850 to uppercase


#---- english version x00-x5F = x00-x5F x60-x6F = x60,x41,x42,x43,x44,x45,x46,x47,x48,x49,x4A,x4B,x4C,x4D,x4E,x4F
x70-x7F = x50,x51,x52,x53,x54,x55,x56,x57,x58,x59,x5A,x7B,x7C,x7D,x7E,x7F x80-xFF
= x80-xFF

#---- german version #x00-x5F = x00-x5F #x60-x6F
= x60,x41,x42,x43,x44,x45,x46,x47,x48,x49,x4A,x4B,x4C,x4D,x4E,x4F #x70-x7F = x50,x51,x52,x53,x54,x55,x56,x57,x58,x59,x5A,x7B,x7C,x7D,x7E,x7F
#x80-x8F = x80,x9A,x82,x83,x8E,x85,x86,x87,x88,x89,x8A,x8B,x8C,x8D,x8E,x8F #x90-x9F
= x90,x91,x92,x93,x99,x95,x96,x97,x98,x99,x9A,x9B,x9C,x9D,x9E,x9F #xA0-xFF = xA0-xFF

[IBMCP_850->UPPER-END] #-------------------------------------------------------------

#-----------------------------------------------------------------------
[USASCII->LOWER] #translate USASCII to lowercase


x00-x3F = x00-x3F x40-x4F = x40,x61,x62,x63,x64,x65,x66,x67,x68,x69,x6A,x6B,x6C,x6D,x6E,x6F
x50-x5F = x70,x71,x72,x73,x74,x75,x76,x77,x78,x79,x7A,x5B,x5C,x5D,x5E,x5F x60-xFF
= x60-xFF
```

```
[USASCII->LOWER-END] #---------------------------------------------------------

#---------------------------------------------------------------
[USASCII->UPPER] #translate USASCII to uppercase


x00-x5F = x00-x5F x60-x6F = x60,x41,x42,x43,x44,x45,x46,x47,x48,x49,x4A,x4B,x4C,x4D,x4E,x4F
x70-x7F = x50,x51,x52,x53,x54,x55,x56,x57,x58,x59,x5A,x7B,x7C,x7D,x7E,x7F x80-xFF
= x80-xFF

[USASCII->UPPER-END] #---------------------------------------------------------

[CASE-TRANSLATION-END] #********************************************************

#****************************************************************
[IDENTIFIER-VALIDATION]

# tables for validation of
identifiers, object names and library names. # # type of name first character
subsequent character # checked against checked against # --------------------------------------------------------------------
# identifier FIRST-CHAR * SUBSEQUENT-CHAR # *) ’&’ only allowed for dynamic
source generation # object name FIRST-CHAR * SUBSEQUENT-CHAR # *) ’&’ not
allowed # library name LIB-FIRST-CHAR LIB-SUBSEQUENT-CHAR # # The tables listed
below define valid identifiers, object and library names # as described in the
Natural manuals. # If you want to handle additional/other characters e.g. language
specific # characters you have to modify the tables accordingly.


#-------------------------------------------------------------- [ISO8859_1]

# special characters # # non DB variable # ’#’ NON-DB-VARI
= x23

# SQL separator character for null/length indicator
# ’@’ SQL-SEP-CHAR = x40 # dynamic source generation # ’&’ DYNAMIC-SOURCE
= x26 # global variable # ’+’ GLOBAL-VARI = x2B

# valid
first characters # ’#’ ’&’ ’+’ ’A’-’Z’ ’a’-’z’ FIRST-CHAR = x23,x26,x2B,x41-x5A,x61-x7A

# valid subsequent characters # ’#’ ’$’ ’&’ ’-’
’/’ ’0’-’9’ SUBSEQUENT-CHAR = x23,x24,x26,x2D,x2F,x30-x39


# ’@’ ’A’-’Z’ ’_’ ’a’-’z’ SUBSEQUENT-CHAR-1 = x40,x41-x5A,x5F,x61-x7A


# valid first characters for library names # ’A’-’Z’ ’a’-’z’ LIB-FIRST-CHAR =
x41-x5A,x61-x7A

# valid subsequent characters for library
names # ’-’ ’0’-’9’ ’A’-’Z’ ’_’ ’a’-’z’ LIB-SUBSEQUENT-CHAR = x2D,x30-x39,x41-x5A,x5F,x61-x7A

# alternate symbol for ’^’ # input blank for IC, LC,
TC and edit masks, # not symbol for ciompare operators like ^= , ^< and ^>
# ’^’ ALTERNATE-CARET = xAA

[ISO8859_1-END] #------------------------------------------------------------

#---------------------------------------------------------------
[IBMCP_850]

# special characters # # non DB variable
# ’#’ NON-DB-VARI = x23
# SQL separator character for
null/length indicator # ’@’ SQL-SEP-CHAR = x40 # dynamic source generation # ’&’
DYNAMIC-SOURCE = x26 # global variable # ’+’ GLOBAL-VARI = x2B

# valid first characters # ’#’ ’&’ ’+’ ’A’-’Z’ ’a’-’z’ FIRST-CHAR = x23,x26,x2B,x41-x5A,x61-x7A

# valid subsequent characters # ’#’ ’$’ ’&’ ’-’
’/’ ’0’-’9’ SUBSEQUENT-CHAR = x23,x24,x26,x2D,x2F,x30-x39

# ’@’ ’A’-’Z’ ’_’ ’a’-’z’ SUBSEQUENT-CHAR-1 = x40,x41-x5A,x5F,x61-x7A

# valid first characters for library names # ’A’-’Z’ ’a’-’z’ LIB-FIRST-CHAR =
x41-x5A,x61-x7A

# valid subsequent characters for library
names # ’-’ ’0’-’9’ ’A’-’Z’ ’_’ ’a’-’z’ LIB-SUBSEQUENT-CHAR = x2D,x30-x39,x41-x5A,x5F,x61-x7A

# alternate symbol for ’^’ # input blank for IC, LC,
TC and edit masks, # not symbol for ciompare operators like ^= , ^< and ^>
# ’^’ ALTERNATE-CARET = xAA

[IBMCP_850-END] #---------------------------------------------------------------------

#---------------------------------------------------------------------
[USASCII]

# special characters # # non DB variable
# ’#’ NON-DB-VARI = x23

# SQL separator character for
null/length indicator # ’@’ SQL-SEP-CHAR = x40 # dynamic source generation # ’&’
DYNAMIC-SOURCE = x26 # global variable # ’+’ GLOBAL-VARI = x2B

# valid first characters # ’#’ ’&’ ’+’ ’A’-’Z’ ’a’-’z’ FIRST-CHAR = x23,x26,x2B,x41-x5A,x61-x7A

# valid subsequent characters # ’#’ ’$’ ’&’ ’-’
’/’ ’0’-’9’ SUBSEQUENT-CHAR = x23,x24,x26,x2D,x2F,x30-x39

# ’@’ ’A’-’Z’ ’_’ ’a’-’z’ SUBSEQUENT-CHAR-1 = x40,x41-x5A,x5F,x61-x7A

# valid first characters for library names # ’A’-’Z’ ’a’-’z’ LIB-FIRST-CHAR =
x41-x5A,x61-x7A

# valid subsequent characters for library
names # ’-’ ’0’-’9’ ’A’-’Z’ ’_’ ’a’-’z’ LIB-SUBSEQUENT-CHAR = x2D,x30-x39,x41-x5A,x5F,x61-x7A

# alternate symbol for ’^’ # input blank for IC, LC,
TC and edit masks, # not symbol for ciompare operators like ^= , ^< and ^>
# ’^’ ALTERNATE-CARET = x5E

[USASCII-END] #-----------------------------------------------------------------------

[IDENTIFIER-VALIDATION-END] #*********************************************************************

#*********************************************************************
[CHARACTER-CLASSIFICATION]
```

```
# classification of characters
according to the C library functions # # isascii # isalpha # isalnum # isdigit
# isxdigit # islower # isupper # iscntrl # isprint # ispunct # isgraph # isspace
# # The classification tables are valid for the English language. # If you want
to handle language specific characters you have to modify the # following tables
accordingly. # The required modifications for the German language are given as
comment.

#-----------------------------------------------------------------------
[ISO8859_1] # valid for ISO8859_1 as internal character set # ASCII characters
ISASCII = x00-xFF

# alphabetical characters # ’A’-’Z’
’a’-’z’ # #---- english version ISALPHA = x41-x5A,x61-x7A #---- german version
AE OE UE ss ae oe ue #ISALPHA = x41-x5A,x61-x7A,xC4,xD6,xDC,xDF,xE4,xF6,xFC

# alphanumeric characters # ’0’-’9’ ’A’-’Z’ ’a’-’z’ # #----
english version ISALNUM = x30-x39,x41-x5A,x61-x7A #---- german version AE OE UE
ss ae oe ue #ISALNUM = x30-x39,x41-x5A,x61-x7A,xC4,xD6,xDC,xDF,xE4,xF6,xFC

# digit characters # ’0’-’9’ ISDIGIT = x30-x39


# hexadecimal digit characters # ’0’-’9’ ’A’-’F’ ’a’-’f’ ISXDIGIT = x30-x39,x41-x46,x61-x66

# lowercase characters # ’a’-’z’ # #---- english version
ISLOWER = x61-x7A #---- german version ss ae oe ue #ISLOWER = x61-x7A,xDF,xE4,xF6,xFC

# uppercase characters # ’A’-’Z’ # #---- english version
ISUPPER = x41-x5A #---- german version AE OE UE #ISUPPER = x41-x5A,xC4,xD6,xDC

# control characters # ISCNTRL = x00-x1F,x7F-x9F

# printable characters # ISPRINT = x20-x7E,xA0-xFF

# special characters # ISPUNCT = x21-x2F,x3A-x40,x5B-x60,x7B-x7E,xA1-xBF,xD7,xF7

# graphical characters # ISGRAPH = x21-x7E,xA1-xFF

# spacing characters # ISSPACE = x09-x0D,x20

[ISO8859_1-END] #-----------------------------------------------------------------

#----------------------------------------------------------------------
[IBMCP_850] # valid for IBMCP_850 as internal character set # ASCII characters
ISASCII = x00-xFF

# alphabetical characters # ’A’-’Z’
’a’-’z’ # #---- english version ISALPHA = x41-x5A,x61-x7A #---- german version
ue ae AE oe OE UE ss #ISALPHA = x41-x5A,x61-x7A,x81,x84,x8E,x94,x99,x9A,xE1

# alphanumeric characters # ’0’-’9’ ’A’-’Z’ ’a’-’z’ # #----
english version ISALNUM = x30-x39,x41-x5A,x61-x7A #---- german version ue ae AE
oe OE UE ss #ISALNUM = x30-x39,x41-x5A,x61-x7A,x81,x84,x8E,x94,x99,x9A,xE1

# digit characters # ’0’-’9’ ISDIGIT = x30-x39


# hexadecimal digit characters # ’0’-’9’ ’A’-’F’ ’a’-’f’ ISXDIGIT = x30-x39,x41-x46,x61-x66

# lowercase characters # ’a’-’z’ # #---- english version
ISLOWER = x61-x7A #---- german version ue ae oe ss #ISLOWER = x61-x7A,x81,x84,x94,xE1
```

```
# uppercase characters # ’A’-’Z’ # #---- english version
ISUPPER = x41-x5A #---- german version AE OE UE #ISUPPER = x41-x5A,x8E,x99,x9A

# control characters # ISCNTRL = x00-x1F,xFF

# printable characters # ISPRINT = x20-x7E,x80-xFE

# special characters # ISPUNCT = x21-x2F,x3A-x40,x5B-x60,x7B-x7E,x9C,x9E,x9F,xA6-xB4,xB8-xC5
ISPUNCT-1 = xC8-xCF,xD5,xD9-xDD,xDF,xE6,xEE-xFE

# graphical
characters # ISGRAPH = x21-x7E,x80-xFE

# spacing characters
# ISSPACE = x09-x0D,x20

[IBMCP_850-END] #--------------------------------------------------------------------

#-----------------------------------------------------------------------
[USASCII] # valid for USASCII as internal character set # ASCII characters ISASCII
= x00-x7F

# alphabetical characters # ’A’-’Z’ ’a’-’z’
# ISALPHA = x41-x5A,x61-x7A

# alphanumeric characters
# ’0’-’9’ ’A’-’Z’ ’a’-’z’ # ISALNUM = x30-x39,x41-x5A,x61-x7A


# digit characters # ’0’-’9’ ISDIGIT = x30-x39

# hexadecimal
digit characters # ’0’-’9’ ’A’-’F’ ’a’-’f’ ISXDIGIT = x30-x39,x41-x46,x61-x66

# lowercase characters # ’a’-’z’ # ISLOWER = x61-x7A

# uppercase characters # ’A’-’Z’ # ISUPPER = x41-x5A

# control characters # ISCNTRL = x00-x1F,x7F

# printable characters # ISPRINT = x20-x7E


# special characters # ISPUNCT = x21-x2F,x3A-x40,x5B-x60,x7B-x7E


# graphical characters # ISGRAPH = x21-x7E

# spacing
characters # ISSPACE = x09-x0D,x20

[USASCII-END] #----------------------------------------------------------------------

[CHARACTER-CLASSIFICATION-END] #*************************************************************************
```